

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

VIRTAMOVE, CORP.,

Plaintiff,
v.

HEWLETT PACKARD ENTERPRISE
COMPANY,

Defendant.

Case No. 2:24-cv-00093-JRG
(Lead Case)

JURY TRIAL DEMANDED

VIRTAMOVE, CORP.,

Plaintiff,
v.

INTERNATIONAL BUSINESS MACHINES
CORP.,

Defendant.

Case No. 2:24-CV-00064-JRG
(Member Case)

JURY TRIAL DEMANDED

**PLAINTIFF VIRTAMOVE'S OPENING
CLAIM CONSTRUCTION BRIEF**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	Claim terms for the '814 Patent.....	1
A.	“disparate computing environments” (claim 1)	1
B.	“system files” (claims 1, 10).....	3
III.	Claim terms for the '058 Patent.....	5
A.	“critical system elements” / “operating system critical system elements” / “Shared library critical system elements” (claim 1)	5
B.	“functional replicas” (claim 1).....	8
C.	“forms a part of the one or more of the plurality of software applications” (claim 1)	10
D.	“shared library” (claim 1)	11
IV.	CONCLUSION.....	13

TABLE OF AUTHORITIES

Cases

<i>Interval Licensing LLC v. AOL, Inc.</i> , 766 F.3d 1364 (Fed. Cir. 2014).....	8
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	1
<i>Sonix Tech. Co. v. Publications Int’l, Ltd.</i> , 844 F.3d 1370 (Fed. Cir. 2017).....	8
<i>Union Paper-Bag Mach. Co. v. Murphy</i> , 97 U.S. 120 (1877).....	9
<i>VirtaMove v. Amazon.com</i> , Case No. 7:24-cv-00030-ADA-DTG (W.D. Tex.)	2
<i>VirtaMove v. Google</i> , Case No. 7:24-cv-00033-DC-DTG (W.D. Tex.)	2
<i>Warner-Jenkinson Co. v. Hilton Davis Chem. Co.</i> , 520 U.S. 17 (1997).....	9

I. INTRODUCTION

Regarding Defendants’ allegations of indefiniteness, Defendants’ position rests on the flawed premise that a POSITA would evaluate the claim terms divorced from the asserted claims, directly contrary to Federal Circuit precedent that claim context is useful in discerning the meaning of claim terms. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (“To begin with, the context in which a term is used in the asserted claim can be highly instructive.”).

Regarding other disputes, Defendants pick and choose quotations from the specification or intrinsic record and propose redefining claim language, even when their proposals are confusing or misleading in the full claim context (and in some cases, are directly inconsistent with the express teachings of the specification). VirtaMove’s proposed constructions seek to clarify claim scope for a factfinder, while still remaining consistent with the intrinsic record.

II. Claim terms for the ’814 Patent.

A. “disparate computing environments” (claim 1)

VirtaMove’s Proposed Construction	Defendants’ Proposed Construction
environments run by standalone computers	environments where computers are stand-alone or where there are multiple computers and where they are unrelated

Claim 1 of U.S. Patent No. 7,519,814 (Ex. 1) recites, in part, a method practiced “[i]n a system having a plurality of servers with operating systems that differ, operating in disparate computing environments.” It is clear from the context of the claim language itself that these “disparate computing environments” are environments that run on a computer and will allow for differing operating systems to operate.

Regarding what additional is required of the claims, the specification makes clear that computing environments are *disparate* “where computers are stand-alone or where there are plural computers and where they are unrelated.” ’814 Patent, 2:17-19. In other words, if two different

environments are run by stand-alone or unrelated computers, those environments are “disparate” from one another. In contrast, if two different environments are run by the same computer or related computers, those two environments are *not* “disparate.”

VirtaMove’s proposed construction omits the “unrelated” aspect of this understanding, because it has been generally agreed that computers that are part of the same system (as required by claim 1) would not be “unrelated,” such that in the context of claim 1 as a whole, “disparate computing environments” could only be satisfied by environments run by standalone computers. *See* Ex. 2 (Amazon’s Opening Brief in *VirtaMove v. Amazon.com*, Case No. 7:24-cv-00030-ADA-DTG (W.D. Tex.)) at 4 (“A POSITA would understand that, for multiple servers to form a single system, they must somehow be related to each other (e.g., networked or otherwise connected).”); Ex. 3 (Google’s Opening Brief in *VirtaMove v. Google*, Case No. 7:24-cv-00033-DC-DTG (W.D. Tex.)) at 6-7 (“But as a matter of common sense, servers in the same system would seem to be ‘related.’”); Ex. 4 (Google’s Reply Brief in *VirtaMove v. Google*) at 2 (“And the claim as a whole *undisputably* cannot extend to ‘unrelated’ computers.” (emphasis in original; internal quotations omitted)).

While Defendants’ proposed construction touches on these similar principles, it leads to absurd results when inserted directly into the claim language. With Defendants’ construction, the claim would recite a method practiced “[i]n a system having a plurality of servers with operating systems that differ, operating in *environments where computers are stand-alone or where there are multiple computers and where they are unrelated.*” But taken literally, this would render the limitation effectively meaningless, because the “Solar System” is an “environment where computers are standalone” and also an environment “where there are multiple computers and where they are unrelated.” And all plausible implementations of the claim would operate within

the Solar System environment, and would thus meet the “disparate computing environments” limitation under a literal insertion of Defendants’ construction into the claim language. Of course, this is not how a POSITA would understand “disparate computing environments” to be applied in the context of the claim language and specification, but this is precisely what Defendants’ proposed construction would imply to a jury.

VirtaMove’s proposed construction clarifies that “environment” in this context relates not to the typical lay understanding of environment (which could include a house, a state, or a planet), but rather the ordinary understanding of a *computing* environment where that environment is *running on a computer*. The specification’s discussion of “disparate computing environments” as being “[e]nvironments where computers are stand-alone” was drafted with this understanding in mind—“where” is a reference to where the environment is running, not what the environment contains. VirtaMove’s construction thus clarifies this concept to a jury, and avoids the potential confusion that Defendants’ construction would cause when inserted directly into the claim language.

B. “system files” (claims 1, 10)

VirtaMove’s Proposed Construction	Defendants’ Proposed Construction
No construction necessary; plain and ordinary meaning.	files provided with an operating system and which are available to applications as shared libraries and configuration files

“System files” have a plain meaning—they are operating system files that include (but are not limited to) shared library files and configuration files. *See* ’814 Patent at 3:14-19 (teaching that in addition to “shared library files and configuration files,” “[t]here may be any number of other files included as system files”). Thus, either a shared library file and a configuration file can be a “system file.”

While Defendants’ proposed construction touches on this concept, it introduces ambiguity by defining “system files” as a *collective*, rather than defining what *individual* system files are. This invites confusion over the meaning of “and”: does each system file need to be both a shared library and a configuration file? This is like defining “pets” as “domestic animals which are cats and dogs.” The danger is that the jury may interpret this definition as requiring that each (individual) system file must meet the requirement of the proposed construction which addresses “system files” (collectively). For instance, the jury may think that if a file is provided with an operating system, but that file is only available to applications as a configuration file, it cannot be a “system file” because it is not *also* available to applications as a shared library.

Notably, the specification is explicit that each individual “system file” *need not* be available as *both* a shared library *and* a configuration file. For instance, the specification teaches that “Linux Apache uses *the following shared libraries*, supplied by the OS distribution, *which are ‘system’ files.*” ’814 Patent at 2:55-3:5 (emphasis added). The specification makes clear that these libraries are distinct from “configuration files,” which are listed separately. *Id.*, 3:6-13. In other words, the shared library files are system files, even though they are not each “available to applications as shared libraries *and* configuration files.”

Defendants’ proposed construction is further improper because it implies that system files are limited to only “shared library” and “configuration files.” But the specification is clear that “system files” is need not consist *only of* “shared library files and configuration files,” when it teaches that that “[t]here may be any number of *other* files included as system files,” which “might be included, for example, to support maintenance activities or to start other network services to be associated with a container.” *Id.* at 3:14-19. Accordingly, Defendants’ attempt to restrict “system files” to *only* shared library and configuration files should be rejected.

Notably, Defendants have not provided any evidence that “system files” lack a plain meaning such that they need an explicit definition, or that the specification’s discussions of “system files” are somehow inconsistent with the plain meaning of that term. However, if the Court is inclined to accept Defendants’ proposal to restrict “system files” to only shared library and configuration files, VirtaMove proposes that “system files” be construed as “files provided with an operating system, each of which is available to an application as a shared library file or configuration file.”

III. Claim terms for the ’058 Patent.

- A. “critical system elements” / “operating system critical system elements” / “Shared library critical system elements” (claim 1)

VirtaMove’s Proposed Construction	Defendants’ Proposed Construction
For “critical system elements”: “any service or part of a service, ‘normally’ supplied by an operating system, that is critical to the operation of a software application.”	Indefinite

U.S. Patent No. 7,784,058 (Ex. 5) provides an unambiguous definition of the phrase “critical system element[s],” stating what a CSE *is*, not merely providing examples or embodiments. The definition has two basic requirements: first, the CSE is “‘normally’ supplied by an operating system”; second, it is “critical to the operation of a software application.”

Regarding the former requirement, the patent specification provides further context, explaining: “It is traditionally the task of an operating system to provide mechanisms to safely and effectively control access to shared resources. In some instances the centralized control of elements, critical to software applications, hereafter called critical system elements (CSEs)[.] creates a limitation caused by conflicts for shared resources.” ’058 Patent at 1:22-27. This illustrates the conventional arrangement wherein CSEs are “normally” provided by an operating

system (i.e., they are provided by the operating system if the structure of the operating system is not modified beyond its default operation). The specification also provides contrasting examples of the “invention,” consistent with the claims, where “some system elements that are critical to the operation of a software application *are replicated from kernel mode, into user mode....* These system elements are contained in a shared library.” *Id.* at 9:15-19 (emphasis added). The specification parallels the claim requirements and confirms that the OSCSEs recited in limitation 1(b) generally correspond to the operation of a conventional system (where the operating system provides the critical system elements), whereas the SLCSEs of limitation 1(c) generally correspond to a non-conventional aspect of the claimed invention (where the critical system elements are stored in a shared library, outside of the operating system).

Defendants provide a declaration of Dr. Stavrou (Ex. 6) that alleges indefiniteness on two separate bases. First, that a POSITA would allegedly be unable to determine whether a service is “normally supplied by an operating system.” And second, that a POSITA would allegedly be unable to determine whether a service is “critical to the operation of a software applications.” Each of these allegations fails for the reasons set forth below.

First, Dr. Stavrou contends that a POSITA would supposedly be unable to determine “whether a service is ‘normally supplied’ by an operating system” because that “differs from one operating system to another.” Ex. 6 ¶ 39. But Dr. Stavrou’s opinion in this regard ignores the fact that the claim recites a “computing system... comprising... *an operating system*,” and then further recites critical system elements. In this context, it is apparent that whether a service is *normally* supplied by “an operating system” would be referring to the *same* “operating system” recited in claim 1. In other words, as long as the service is normally supplied by the *claimed* “an operating system,” it is “normally supplied by an operating system” within the context of claim 1.

And under this common sense understanding of the claim language, Dr. Stavrou provides no coherent theory that a POSITA would be unable to understand whether services are “normally” supplied by the particular operating system within utilized within the “system” of claim 1. Indeed, Dr. Stavrou’s opinion is that “[a] POSITA would not have been aware of any objective standards to determine whether a ‘service or parts of a service’ was ‘normally supplied’ *without knowing the particular version of the operating system in question*” (*id.* ¶ 40 (emphasis added)), thus acknowledging that a POSITA *would* be able to determine whether a service was “normally supplied” with respect to a *specific* operating system as the claims require.

Second, Dr. Stavrou contends that “[w]hether a service is ‘critical’ to the operating of a software application is subjective, and will depend on the specific context of the software application, including its specific functionalities,” and would “also depend[] on the operating system in question.” *Id.* ¶¶ 35-36; *see also id.* ¶ 38 (“Thus, the ‘critical’ quality of any service would not only depend on the software applications a POSITA may consider, but also on the operating systems, the universe of which is not defined by the ’058 Patent.”).

Dr. Stavrou’s opinions do not support indefiniteness of this term, because they again fail to consider that the *claim* recites both “an operating system” and “a plurality of software applications.” ’058 Patent at 10:51-59. Thus, the “software applications” and “the operating system” *is* defined in the context of claim 1—it is the operating system and software applications mapped to the “operating system” and “software applications” of claim 1. Dr. Stavrou’s acknowledgement that criticality of services *would be* understandable to a POSITA given the context of both the operating system and software applications expressly undermines Defendants’ indefiniteness contention.

B. “functional replicas” (claim 1)

VirtaMove’s Proposed Construction	Defendants’ Proposed Construction
“substantial functional equivalents or replacements of kernel functions”	Indefinite

The Federal Circuit has explained that “[b]ecause language is limited, we have rejected the proposition that claims involving terms of degree are inherently indefinite.” *Sonix Tech. Co. v. Publications Int’l, Ltd.*, 844 F.3d 1370, 1377 (Fed. Cir. 2017). “Thus, a patentee need not define his invention with mathematical precision in order to comply with the definiteness requirement.” *Id.* (internal quotation marks omitted). “Claim language employing terms of degree has long been found definite where it provided enough certainty to one of skill in the art when read in the context of the invention.” *Interval Licensing LLC v. AOL, Inc.*, 766 F.3d 1364, 1370 (Fed. Cir. 2014). In determining whether the patent has provided sufficient guidance for a term of degree, a reviewing court should “look to the written description for guidance.” *Id.* at 1371.

Defendants again rely on Dr. Stavrou’s declaration, which contends that “replica” (but not “functional replica”) has a lexicographic definition, *i.e.* “a CSE having similar attributes to, but not necessarily and preferably not an exact copy of a CSE in the operating system (OS),” and that definition is indefinite as a term of degree because of the term “similar attributes to.” Ex. 6 ¶ 52. This contention fails at both steps.

First, Dr. Stavrou’s primary focus on a single sentence from the patent specification ignores the claim context and the full disclosure of the patent specification. In particular, the claim term is “functional replica,” not “replica.” Even if the generic description of “replica” were indefinite (it is not), the limitation to functional replicas provides important clarification.

The specification contains an additional description of the scope of “the term replica” specifically in the context of *functional* replicas: “The CSE library includes replicas or substantial

functional equivalents or replacements of kernel functions. The term replica, shall encompass any of these meanings, and although not a preferred embodiment, may even be a copy of a CSE that is part of the OS.” ’058 Patent at 8:27-32; *see also id.* at 9:52-56 (“The term replication means that like services are supplied [*i.e.*, that] essentially a same functionality is provided.”). These sentences explicitly state what scope is “encompass[ed]” by “the term replica”: (1) substantial functional equivalents of kernel functions; (2) replacements of kernel functions; and (3) copies of OSCSEs (*i.e.*, kernel functions). Of these three categories, “substantial functional equivalents” is logically the broadest, since either a replacement or a copy of a kernel function/OSCSE would necessarily also be functionally equivalent.

Accordingly, the phrase “functional replica” does not require mere similarity, but rather (at a minimum) “substantial functional equivalen[ce].” ’058 Patent at 8:27-32.

Dr. Stavrou’s discussion of this “substantial functional equivalents” language is both (1) conclusory and (2) inconsistent with inquiries that the Supreme Court assigns to factfinders in evaluating the doctrine of equivalents. Specifically, Dr. Stavrou contends that “the term ‘substantial’ is a subjective term of degree” such that a POSITA would be unable to determine whether a function is a “substantial functional equivalent” of another. Ex. 6 ¶ 59. Dr. Stavrou provides no basis for his opinion, other than that “substantial” is a “term of degree.” But his opinion that “substantial” is indefinite is directly inconsistent with Supreme Court precedent that whether infringement under the doctrine of equivalents is satisfied should be measured by respect to whether the accused product “performs substantially the same function” as recited by a claim. *Warner-Jenkinson Co. v. Hilton Davis Chem. Co.*, 520 U.S. 17, 38 (1997) (quoting *Union Paper-Bag Mach. Co. v. Murphy*, 97 U.S. 120, 125 (1877)). Dr. Stavrou’s argument, if accepted, would

be directly inconsistent with a core premise underlying longstanding Supreme Court precedent—that factfinders *are* able to determine substantial functional equivalence.

Accordingly, Defendants’ argument that “functional replica” (which the patent makes clear extends to the “substantial functional equivalents of a kernel function”—’058 Patent at 8:27-32) is indefinite should be rejected.

C. “forms a part of the one or more of the plurality of software applications” (claim 1)

VirtaMove’s Proposed Construction	Defendants’ Proposed Construction
No construction necessary; plain and ordinary meaning.	literally form a part of the application such that it resides in the same address space as application code, in contrast to a proxy that is exclusive of the application

Defendants do not point to anything in the file history that differs from the plain meaning of “forms a part of the one or more plurality of software applications.” Instead, Defendants seek to introduce redundancy and confusion into the claims on the basis of a prior art distinction.

In prosecution, the applicant made clear that in order for a critical system element to “form a part of the one or more of the plurality of software applications” when it is accessed, it must “reside[] in the same address space as” the application’s code. ’058 Patent File History (Ex. 7) at 32. VirtaMove does not dispute that this is a requirement under the plain meaning of “forms a part of...” (i.e., it would not be possible for a critical system element to form a part of an application if the critical system element did not at a minimum, reside in the same address space as the application code), but that is also not a lexicographical definition but instead just a reinforcement of the plain meaning. Likewise, specifying that to “form a part of [an] application” should be taken “literally,” such that something “that is exclusive of the application” would not satisfy the claim language is not a definition, but just a reinforcement of plain meaning of the literal claim language.

By adding this non-definitional discussion from the prosecution history, Defendants’ construction would place undue emphasis on this limitation by saying it must be “literally” met. Of course, that is true under a literal infringement theory, but that is *also* true of *all* claim elements, and there is no reason to give undue emphasis to this claim element just because the applicant emphasized the literal plain meaning of that term to an examiner during prosecution.

D. “shared library” (claim 1)

VirtaMove’s Proposed Construction	Defendants’ Proposed Construction
“an application library <i>whose</i> code space <i>is</i> shared among all user mode applications”	“an application library code space shared among all user mode applications. The code space is different than that occupied by the kernel and its associated files. The shared library files are placed in an address space that is accessible to multiple applications,” wherein an “application library” is “a collection of functions in an archive format that is combined with an application to export system elements”

The term “shared library” appears throughout the specification and claims of the ’058 Patent. It has a plain and ordinary meaning that is confirmed by the claim context and by the specification. Instead of recognizing this plain and ordinary meaning, Defendants demand including a confusing, circular “definition” that introduces additional terms unfamiliar to a jury and that, appears to contradict the claim context.

For example, the patent specification makes clear that “code space” refers to where a library is located, not to the library itself. *See, e.g.*, ’058 Patent at 3:39-45 (“the same set of instructions in the same physical memory space, *that is, shared code space...*”); *id.* at 6:54-55 (“Static library: An application *whose* code space is *contained* in a single application”); *id.* at 7:3-5 (“[W]hat is commonly done is to provide an application library *in* shared code space, which multiple applications can access.”). This usage, which reflects the plain and ordinary meaning of

“code space” to a POSITA, contradicts the notion that a shared library is *defined* as “an application library code space” as Defendants request.

There is a simple explanation for the confusing construction, though: the patent applicant obviously introduced a pair of typographical errors into the definition of “Shared library.” The original version of this definition, in the provisional application to which the ’058 Patent claims priority, is shorter: “An application library *whose* code space *is* shared among all user mode applications.” Provisional Patent Application No. 60/504,213 (Ex. 8) at 9. That definition cleanly flowed from the definition of “Application library” above it, and paralleled the definition of “Static library” below it, confirming that the key difference between a shared library and a static library is whether the code space is contained in a single application or shared among applications:

Application library: A collection of functions in an archive format that is combined with an application to export system elements.

Shared library: An application library whose code space is shared among all user mode applications.

Static library: An application library whose code space is contained in a single application.

Id.

When the applicant revised the provisional specification to form the non-provisional application, additional detail was added to the definition, but the words “whose” and “is” were removed. Those words were not deleted from the definition of “Static library,” which retains the same definition in the final specification. The new language includes “The code space is different than that occupied by the kernel,” confirming that “code space” is a space occupied by code, not code itself. This confirms that the deletion of “whose” was unintentional, and that the correct

interpretation should retain the original language of the provisional. A POSITA reading the specification would readily understand that this is the correct interpretation. Therefore, if the extent the Court believes construction is necessary, the correct definition without the typographical errors should be included: “An application library *whose* code space *is* shared among all user mode applications.”

IV. CONCLUSION

For the foregoing reasons, VirtaMove respectfully requests that VirtaMove’s proposed constructions be adopted, and Defendants’ indefiniteness allegations be rejected as inconsistent with the evidence of record and common sense.

Dated: February 21, 2025

Respectfully submitted,

/s/ Reza Mirzaie

Reza Mirzaie
CA State Bar No. 246953
Marc A. Fenster
CA State Bar No. 181067
Neil A. Rubin
CA State Bar No. 250761
Jacob R. Buczko
CA State Bar No. 269408
James S. Tsuei
CA State Bar No. 285530
James A. Milkey
CA State Bar No. 281283
Christian W. Conkle
CA State Bar No. 306374
Jonathan Ma
CA State Bar No. 312773
Daniel Kolko
CA State Bar No. 341680
RUSS AUGUST & KABAT
12424 Wilshire Boulevard, 12th Floor
Los Angeles, CA 90025
Telephone: 310-826-7474
Email: rmirzaie@raklaw.com

Email: mfenster@raklaw.com

Email: nrubin@raklaw.com

Email: jbuczko@raklaw.com

Email: jtsuei@raklaw.com

Email: jmilkey@raklaw.com

Email: cconkle@raklaw.com

Email: jma@raklaw.com

Email: dkolko@raklaw.com

Qi (Peter) Tong

TX State Bar No. 24119042

8080 N. Central Expy, Suite 1503

Dallas, TX 75206

Email: ptong@raklaw.com

**ATTORNEYS FOR PLAINTIFF
VIRTAMOVE, CORP.**

CERTIFICATE OF SERVICE

I certify that this document is being served upon counsel of record for Defendants on February 21, 2025 via CM/ECF.

/s/ Reza Mirzaie